

# Progressive Clustering with Learned Seeds: An Event Categorization System for Power Grid

Boyi Xie<sup>†</sup>, Rebecca J. Passonneau<sup>†</sup>, Haimonti Dutta<sup>†</sup>,  
Jing-Yeu Miaw, Axinia Radeva, Ashish Tomar  
Center for Computational Learning Systems  
Columbia University  
New York, USA 10027  
Email: †{xie@cs,becky@cs,haimonti@ccls}.columbia.edu

Cynthia Rudin  
MIT Sloan School of Management  
Massachusetts Institute of Technology  
Cambridge, USA 02139  
Email: rudin@mit.edu

**Abstract**—Advances in computational intelligence provide improved solutions to many challenging software engineering problems. Software has long been deployed for infrastructure management of utilities, such as the electric power grid. System intelligence is in increasing demand for system control and resource allocation. We present a model for electrical event categorization in a power grid system: Progressive Clustering with Learned Seeds (PCLS) – a learning method that provides stable and promising categorization results from a very small labeled data. It benefits from supervision but maximally allows patterns be discovered by the data itself. We find it effectively captures the dynamics of a real world system over time.

## I. INTRODUCTION

Advances in computational intelligence provide improved solutions to many challenging software engineering problems. Software has long been deployed for infrastructure management of utilities, such as the electric power grid or telecommunication systems. System intelligence is in increasing demand for system control and resource allocation. Our work applies machine learning to a problem for the low power electrical grid that directly services customers. Over the past half dozen years, we have worked closely with Consolidated Edison of New York, a major utility company in New York City, on a project to apply machine learning techniques to the secondary electrical grid. The results have been used to maintain the reliability of the secondary electrical grid.

The goal of our project is to develop interpretable models on a year-by-year basis to rank secondary structures (manholes and service boxes) with respect to their vulnerability to a serious event, such as fire or explosion. We use a supervised ranking algorithm, thus have a need for labeled data that indicates which structures are vulnerable in a given year for training our models. The main source of data for labeling the structures consists of Con Edison’s Emergency Control System (ECS) trouble tickets. They document electrical events, such as interruptions of service, and engineers’ efforts to redress any problems. The task we address in this paper is how we apply machine learning to the trouble tickets in order to sort them into those that document serious events on structures, and those that document non-serious events. Serious events result in positive labels on the implicated structures; non-serious events are included along with serious events in the

representation of a structure’s past history. Thus the ability to learn a good ranking model for structures depends on our ticket classification.

In this paper, we describe our approach, Progressive Clustering with Learned Seeds (PCLS) – a learning method adapted to this domain that provides stable and promising categorization results from a very small labeled data set. PCLS benefits from supervised learning but maximally allows patterns be discovered by the data itself. We found that it effectively captures the dynamics of a time-varied real world system.

In this context, the goal of our event categorization system is to classify ECS tickets with respect to whether the reported “trouble” is a) a serious event, b) a low-grade event (a minor interruption or disruption of service), or c) not relevant. The semantics of the three-way classification is somewhat subjective and contingent. Certain events are unequivocally serious, such as manhole explosions. However, many tickets pertain to events that can be serious or not, depending on a wide range of factors. Further, the language in the trouble tickets changes over time. From a small, expert-labeled sample for a single region, we initially hand-crafted a set of classification rules (see section III). We use these to initialize a clustering approach for the earlier years of data. To initialize clusters for later years, learned decision trees from the clusters help seed the clusters.

The issue of gradual shifts in the semantics of document classes is potentially a very general one, thus our approach could apply to many problems where results of previous supervised learning must be adapted to changing contexts. Section II presents related work, followed by motivation (section III). We describe the data sets and the general domain in Section IV. Section V introduces a new semi-supervised approach, Progressive Clustering with Learned Seeds (PCLS). Section VI reviews three learning methods and our evaluation procedure. Section VII presents results of our experiments. At last, section VIII briefly summarizes our contribution.

## II. RELATED WORK

Early work on incremental learning [1], [2] attempted to build learners that could distinguish between noise and change. While they deal with concept learning, they address the same

general problem we face. Utgoff [2] presents an incremental decision tree algorithm that restructures the tree as needed for new instances. Training instances are retained in the tree to facilitate restructuring, thus constituting meta-knowledge, meaning knowledge distinct from what is learned, but which facilitates learning.

There has been much previous work on cluster seeding to address the limitation that iterative clustering techniques (e.g. K-Means and Expectation Maximization (EM)) are sensitive to the choice of initial starting points (seeds). The problem addressed is how to select seed points in the absence of prior knowledge. Kaufman and Rousseeuw [3] propose an elaborate mechanism: the first seed is the instance that is most central in the data; the rest of the representatives are selected by choosing instances that promise to be closer to more of the remaining instances. Pena et al. [4] empirically compare the four initialization methods for the K-Means algorithm and illustrate that the random and Kaufman initializations outperform the other two, since they make K-Means less dependent on the initial choice of seeds. In K-Means++ [5], the random starting points are chosen with specific probabilities: that is, a point  $p$  is chosen as a seed with probability proportional to  $p$ 's contribution to the overall potential. Bradley and Fayyad [6] propose refining the initial seeds by taking into account the modes of the underlying distribution. This refined initial seed enables the iterative algorithm to converge to a better local minimum.

*CLustering through decision Tree construction (CLTrees)* [7] is related to ours in their use of decision trees (supervised learning) for generation of clusters. They partition the data space into data and empty regions at various levels of details. Their method is fundamentally different from ours and do not capture the aspect of incremental learning over time.

### III. MOTIVATION

We have been working with Con Edison to develop a machine learning approach to predict serious events in secondary structures (manholes and service boxes). Our task is to produce for each borough, for a given year, a ranked list of the borough's structures with respect to their vulnerability to a serious event in the near future. The prediction problem is challenging. Only 0.1-5.0% of the tens of thousands of structures per borough experience a serious event each year, depending on borough, year and the definition of serious event. Causes of these rare events, if they can be detected, are often indirect (insulation breakdown), can depend on a complex of factors (number of cables per structure), and develop slowly. Evaluation consists of a blind test of a ranked list against the serious events that occur in the following year, with emphasis on the top of the ranked lists, which are used to prioritize Con Edison repair work.

To label the structures for supervised learning, we rely on the ticket classes described in the introduction. As described in [8], we developed a data mining, inference and learning framework to rank structures. It relies on a supervised bipartite ranking algorithm that emphasizes the top of a ranked list [9].

For any given year, a structure that is identified as the location of a problem in a serious event ticket gets a positive label; all other structures get negative labels. The feature descriptions of the labeled structures also depend heavily on the ticket classes: across boroughs, years and modeling methods, at least half the features in our ranking models represent how many serious or low-grade events a structure has experienced in the recent or remote past.

We had no a priori gold standard for classifying tickets. Based on the intuitions of two domain experts, we initially used a trouble type assigned to tickets by Con Edison dispatchers as the indicator of seriousness. Of the roughly two and a half dozen trouble types we use (the constituency varies somewhat across boroughs), two are unequivocally serious (for explosions and fires), and some are almost never serious (e.g., flickering lights). However, there is one category in particular (smoking manholes) that is both very large and can be serious or not. In previous work [10], we applied corpus annotation methods to elicit a definition by example of the three classes: serious event, low-grade event, and irrelevant. Two experts labeled a carefully designed sample of 171 tickets. That they achieved only modest interannotator agreement on the first pass ( $\kappa=0.49$ ; [11]) indicates that the classes are rather subjective. Based on a second pass involving adjudication among the experts, we developed a rule-based method to classify tickets. We produced a small fixed set rules, along with a large fixed set of regular expressions, to capture generalizations we observed in the hand-labeled data. To test and refine the rules we applied them to large random samples, modifying them based on our judgments of their accuracy. As reported in [10] the rule-based classes improved the ranking results, particularly at the top of the list for the most vulnerable structures (one in every five structures was affected).

Development of the hand-crafted rules (HCR) required approximately 2,500 person hours. While they improved over the use of the assigned trouble type, our goal in the experiments reported here is to boost the improvement further, and to adapt the rules over time and regions. In particular, we need an approach that generalizes over time and space: the different boroughs have different infrastructure and histories, slightly different sublanguages (in the sense of [12]), and we use different subsets of trouble types as data. Rather than adapting the rules manually, which would be costly, we seek an automated method.

While the notions of *relevant* or *serious* events have some generality, the specific realization of the three ticket class changes from borough to borough, and from year to year. This can be illustrated by comparing the discriminative words over time. If we take the ticket classes produced by our hand-crafted rules and compare the list of words that discriminate the classes from year to year, we find that only about half the discriminative words overlap. For relevant versus non-relevant tickets, a comparison of the discriminative vocabulary for each successive pair of years from 2001 to 2005 shows that the average overlap in vocabulary is only 56.27% (sdev=0.05).

#### IV. DATA SOURCES

We have been working with Con Edison Emergency Control System (ECS) tickets from three New York city boroughs: Manhattan, Brooklyn and Bronx. The experiments reported here pertain to Manhattan, which is our primary focus. We use tickets from 1996 through 2006, consisting of 61,730 tickets. The tickets are textual reports of secondary events, such as manhole fires, flickering lights in a building, and so on; they are generated when a customer or city worker calls the ECS line. The ECS tickets in our dataset range in length from 1 to 550 lines, with a similarly wide range of information content. ECS tickets can have multiple entries from different individuals, some of which is free text, some of which is automatically entered. These tickets exhibit the fragmentary language, lack of punctuation, acronyms and special symbols characteristic of trouble tickets from many arenas.

Structures are labeled with respect to a given year, thus we apply automated ticket classification or clustering on a year-by-year basis. We first classify tickets into relevant versus irrelevant events, then classify the tickets in the relevant class into serious versus low-grade events. The serious versus low-grade event classes are highly skewed. Of 61,730 tickets for ten years of Manhattan ECS (relevant trouble types only), about 43.67% represent relevant events, and only 15.92% of these are serious.

#### V. PROGRESSIVE CLUSTERING WITH LEARNED SEEDS

Progressive Clustering with Learned Seeds is a method adopted after consideration of the tradeoffs of supervised and unsupervised approaches. We aimed to minimize the sensitivity of K-Means clustering to the initial seed points by biasing the initial centroids closer to the optimal ones using prior knowledge about the document classes.

##### Procedure 1 Progressive Clustering with Learned Seeds

- 1: Tree path extraction
- 2: Path scoring
- 3: Class contribution calculation
- 4: Seed points retrieval
- 5: K-Means clustering using retrieved seeds

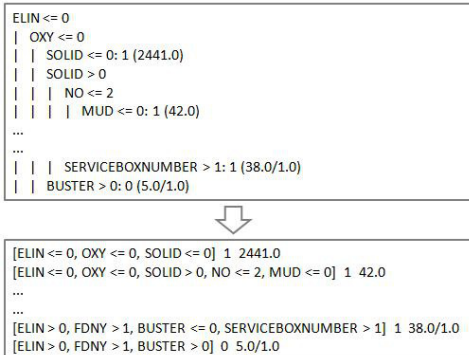


Fig. 1. Tree path extraction, which converts a decision tree model to a collection of paths

#### A. Tree path extraction

We train a decision tree model on the previous year's data, convert it into a set of paths as illustrated in Figure 1, and extract path attributes.

The following attributes are extracted for each path: (1) Length - the number of terms it contains; (2) Coverage - the number of instances addressed; (3) Accuracy - the rate of correctly classified instances; (4) Label - the class it predicts. We assign scores for each path using the first three attributes.

#### B. Path scoring

The scoring process formalizes the intuition that an optimal rule relies on fewer features, has greater coverage and higher accuracy. To score a path, we first compute homogeneous scores for all three attributes, in particular, to make them uniformly distributed within the range  $[0,1]$ . Subsequently, we use coefficients to weight each attribute and calculate a final score, also in  $[0,1]$ .

For path  $i$ , we calculate its  $ScoreLength$ ,  $ScoreCoverage$  and  $ScoreAccuracy$  separately using the following formulas:

$$ScoreLength_i = \frac{l_{max} - l_i}{l_{max} - l_{min}} \quad (1)$$

where  $l_i$  is the length of the path  $i$ , and  $l_{max}$  and  $l_{min}$  are the lengths of the longest and shortest paths.  $ScoreLength_i \in [0, 1]$  is a uniform distribution.

$$ScoreCoverage_i = \frac{CoverageNorm_i}{CoverageNorm_{max}} \quad (2)$$

$$CoverageNorm_i = \log(c_i + 1) \quad (3)$$

where  $c_i$  is the coverage of the path  $i$ , i.e. the number of instances related to path  $i$  in the training data. Because there is a big gap in coverage among a set of paths, e.g. from thousands to only a few, the logarithm function is used to smooth the data into a uniform distribution. By a further normalization,  $ScoreCoverage_i \in [0, 1]$ .

$$ScoreAccuracy_i = \frac{a_i}{a_{max}} \quad (4)$$

where  $a_i$  is the accuracy of the path  $i$ .  $ScoreAccuracy_i \in [0, 1]$ .

In summary, paths that are shorter, have more coverage and are more accurate score higher. After scoring each attribute, we calculate the final score for path  $i$

$$Score_i = \lambda_l \cdot ScoreLength_i + \lambda_c \cdot ScoreCoverage_i + \lambda_a \cdot ScoreAccuracy_i \quad (5)$$

where  $\lambda_l + \lambda_c + \lambda_a = 1$ ;  $\lambda_l$ ,  $\lambda_c$  and  $\lambda_a$  are coefficients for the attribute length, coverage and accuracy respectively. They are used to weight each path attribute in the scoring function. Notice that, because each attribute score is normalized, the final score is also a uniform distribution and  $Score_i \in [0, 1]$ .

### C. Class contribution calculation

Due to the data skew and the differential role of each class in the structure ranking problem, we next rank paths on a per class basis. We assign a quantity to each path representing its relative contribution to the class it predicts. The class contribution for each path exaggerates the discriminative power by an exponential function that increase the larger scores and decreases the smaller ones and is then normalized.

$$ScoreTransform_i = Base^{Score_i} \quad (6)$$

where  $Score_i$  is the score for path  $i$ ,  $Base$  is a base constant of the exponential function.

$ScoreTransform_i$  can be regarded as the raw contribution of path  $i$ . If there are  $N_c$  paths belonging to class  $c$ , the sum of the scores  $\sum_{i=1}^{N_c} ScoreTransform_i$  contributes the whole class. The ratio of  $ScoreTransform_i$  and the sum can be regarded as the contribution of path  $i$ , and is normalized to [0,1]. Where  $c$  is the index for the class:

$$Contribution_{(i,c)} = \frac{ScoreTransform_i}{\sum_{i=1}^{N_c} ScoreTransform_i} \quad (7)$$

Paths are ranked for each class separately by their contribution. Seed points will be selected according to each class's path rank list.

### D. Seed points retrieval

For seed points retrieval, we prefer to choose paths with a higher class contribution, and selects a reasonable number of data for each class from the decision tree model. Given a per class ranking of paths, we use the number of decision tree paths for each class to determine the proportion of seeds for each class, which reflects the relative importance of the information we have learned from the decision tree model.

$$NumOfSeeds_c = P \cdot N_{total} \cdot \frac{count(path_i, c)}{count(path_i)} \quad (8)$$

where  $N_{total}$  is the total number of instances in the data set that we want to cluster,  $P$  is the percentage of data to be seed points,  $count(path_i, c)$  is the number of paths related to class  $c$  and  $count(path_i)$  is the total number of paths extracted from the decision tree model.

When using a path from the tree trained in the prior year to retrieve instances in the current year, there may be no instances, or their may be more than needed. In the latter case, we randomly select the desired number.

### E. Clustering with Learned Seed points

For the initial centroids, we hope to minimize

$$\Theta = \sum_{k=1}^K \|\vec{c}_{init_k} - \vec{c}_{opt_k}\| \quad (9)$$

where  $\vec{c}_{init_k}$  is the initial centroids and  $\vec{c}_{opt_k}$  is the optimal centroids.

Since the classification precision is usually high and stable (see Section VII), by utilizing the paths selected from the decision tree we can select the initial centroids to be more appropriately located in the overall space. Before the K-Means optimization procedure, we initialize the cluster centroid using the seed points we retrieved.

$$\vec{c}_{init_k} = \vec{\mu}_k = \frac{\sum_{n=1}^{N_k} \vec{x}_n}{N_k} \quad (10)$$

where  $\vec{x}_n$  is the  $n^{th}$  instance selected by paths that belong to class  $k$ .  $N_k$  is the total number of instances that were found related to class  $k$ .

We select initial centroids  $\vec{\mu}_k$  to seed the clusters, then apply K-Means.

## VI. METHODS

We seek an automated or semi-automated approach that can classify tickets for the structure ranking task, with adaptation to each borough and time frame. To reiterate, our goal is to improve upon the Hand-Crafted Rule (HCR), thus we use their output as a baseline. Then we compare three learning methods: (1) C4.5 Decision Tree (DT), (2) K-Means Clustering (KM), (3) Progressive Clustering with Learned Seeds (PCLS). In this section, we first introduce our data representation and feature selection method. Next we briefly contrast the strengths and weaknesses of DT and KM; PCLS was described in section V. Then we describe our evaluation method.

### A. Data preprocessing

We use *bag-of-words* document representation, with feature selection to reduce dimensionality. There are an estimated 7,500 distinct unigrams in each year's data, not counting misspellings and word fragments; we use about 10% (750 terms) as features. Previous experiments with spelling normalization reduced the vocabulary by 40%, but had an inconsistent and modest impact. In the experiments reported here, we filter out line separators and other lines with little or no text.

Feature selection was the same for all three classification approaches and was always performed on data from prior year(s). We compared the performance of Bi-Normal Separation, Chi-Square, F-Measure and Information Gain [13] for feature selection. Information Gain exhibited the most stable and consistent performance across different boroughs, years and data representation formats (Boolean, TFIDF, TF). The results reported here all rely on Information Gain for feature selection, and absolute term frequency (TF) as the bag-of-words vector values.<sup>1</sup>

### B. Baseline: Hand Crafted Rules

The hand-crafted rules rely on three types of information: other Con Edison databases indicating the voltage; global properties of the ticket such as length and ticket trouble type;

<sup>1</sup>Absolute TF performs better than normalized TF, presumably because it indirectly represents the length of the ticket, a factor in determining seriousness.

meta-data we assign to indicate signs of seriousness or type of work performed, based on pattern-matching for terms in the ticket. There is only one set of hand-crafted rules that was bootstrapped from a small labeled dataset and it is used for all year’s data.

### C. Decision Trees and Clustering

We used the Weka [14] implementation of the C4.5 decision tree [15] for an interpretable, supervised approach to our classification tasks. In general, the decision tree models exhibited good precision, but with poor recall on serious events, which had a negative effect on structure ranking in that too few structures were labeled as serious. Decision trees are relatively interpretable in comparison to other learning methods because the paths in the tree can be converted to rules for each class being learned. In contrast, the strengths of K-means clustering are speed, a lack of dependence on labeled training data, and high recall. The weaknesses are poor precision, and lack of robust performance due to the sensitivity to initial centroids.

### D. Evaluation

To evaluate the performance of DT, KM and PCLS, we performed intrinsic and extrinsic evaluations [16]. The intrinsic evaluation is to compare the predicted event labels with labels generated by HCR, as measured by recall, precision and F-measure. In the context of our project, the event labeling is in the service of the structure ranking problem and has a crucial impact. We are therefore able to perform an extrinsic evaluation on the structure ranking task. To reiterate, events classified as serious in the year for training the ranking model determine which structures are labeled as vulnerable. Consequently, the extrinsic evaluation provides the most compelling evidence for the merit of the event categorization. Our extrinsic evaluation consists in generating a distinct set of structure labels and features for each event classification method, and comparing the ranked lists that the ranking model yields when relying on each event categorization method.

## VII. EXPERIMENTAL FRAMEWORK AND RESULTS

Our goal is to bootstrap from the rule-based method at some point in the past, then to rely solely on the automated methods from that point forward. We use PCLS, where we cluster the current year  $Y_i$  of tickets, seeding the clusters with seed points selected using the learned trees from the prior year  $Y_{i-1}$ , then in the subsequent year  $Y_{i+1}$ , apply the decision tree of  $Y_i$  for seeding clusters for year  $Y_{i+1}$ .

We report intrinsic and extrinsic evaluation results to compare ticket classes produced by HCR, C4.5 decision trees, KM and PCLS. Figure 2 schematically represents the experimental setup: bootstrap automated classification from HCR in 2001, then evaluate automated methods for ticket classification for 2002-2006. Intrinsic evaluation applies to each year. We report extrinsic results for two ranked lists, for the years 2006 and 2007: the ranked list trained on 2005 data is given a blind evaluation against 2006 data, and the ranked list trained on 2006 is evaluated against 2007 data. The ranking models are

trained using ticket classes to label structures for the training year, and features based on ticket classes for all prior years.

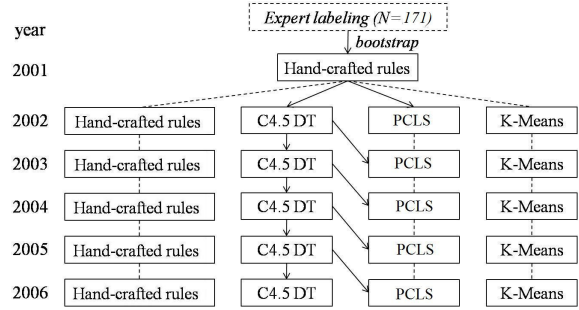


Fig. 2. Experiment setup. We compare Progressive Clustering with Learned Seeds with hand-crafted rules, C4.5 decision tree, and K-Means methods.

### A. Intrinsic results

For PCLS, results shown here use 50% of total instances as seed points to generate initial centroids, the weights of attributes are  $\lambda_l = 0.1$ ,  $\lambda_c = 0.2$  and  $\lambda_a = 0.7$  for path scoring, and  $Base = 2^{110}$  for class contribution. The results in Table I show that PCLS dramatically improves over K-means with random seeding for the serious versus low-grade event classification task; average F-measure for both classes is always larger for PCLS (significantly better for 4 out of 5 years). Compared with the C4.5 decision tree, PCLS exhibits a better recall and F-measure on the serious class (each is better for 4 out of 5 years, and far better for 2005 and 2006) while maintaining a competitive overall performance (in particular, better F-measure for 2004-2006). For the classification of relevant versus irrelevant events, we achieve similar results but do not present them here due to space limitations. Naive Bayes (NB) and SVM classifiers are also experimented. NB has worse performance. SVM achieves similar results but provides less human interpretable model than DT, such as the criterion of tree node. Thus, NB and SVM results are not reported here.

### B. Extrinsic results

To rank structures, we use a bipartite ranking algorithm that focuses on the top of a ranked list [9]. For evaluation, we report AUC, DCG (Discounted Cumulative Gain, a weighted version of AUC that favors the top of the list) and PNorm scores, as recommended in [9]. A blind evaluation assesses how well the ranked list predicts structures that had particularly serious events in the year following the training year, where we have no access to the ECS tickets.

Measure	HCR	DT	KM	PCLS
AUC	0.524	0.516	0.540	<b>0.560</b>
DCG	30.789	30.625	31.447	<b>31.834</b>
PNorm	1.24E+09	1.29E+09	1.15E+09	<b>1.06E+09</b>

TABLE II  
EXTRINSIC EVALUATION BY AUC, DCG AND PNORM MEASURES.  
HIGHER SCORE IS PREFERRED FOR AUC AND DCG, AND LOWER SCORE  
IS PREFERRED FOR PNORM.



year	class	C4.5 Decision Tree (DT)				K-Means Clustering (KM)				PCLS			
		Pre	Rec	F	Avg. F	Pre	Rec	F	Avg. F	Pre	Rec	F	Avg. F
2002	serious event	.701	.603	.648	.746	.024	.162	.042	.366	.610	.716	.659	<b>.718</b>
	low-grade event	.835	.852	.843		.757	.634	.690		.879	.698	.778	
2003	serious event	.675	.491	.569	.668	.256	.536	.347	.319	.328	.428	.371	<b>.497</b>
	low-grade event	.813	.725	.767		.908	.174	.292		.792	.514	.623	
2004	serious event	.633	.376	.472	.599	.251	.577	.350	.429	.688	.414	.517	<b>.606</b>
	low-grade event	.706	.745	.725		.631	.425	.508		.690	.701	.695	
2005	serious event	.536	.635	.581	.657	.624	.895	.735	.688	.560	.932	.700	<b>.696</b>
	low-grade event	.780	.692	.733		.731	.571	.641		.691	.694	.693	
2006	serious event	.603	.554	.577	.646	.008	.128	.016	.320	.692	.791	.738	<b>.712</b>
	low-grade event	.708	.722	.715		.547	.726	.624		.710	.664	.686	

TABLE I

INTRINSIC EVALUATION OF SERIOUS VERSUS LOW-GRADE EVENT CATEGORIZATION. THE RESULTS FOR C4.5 DECISION TREE, K-MEANS CLUSTERING AND PCLS ARE COMPARED WITH THE LABELS FROM HAND-CRAFTED RULES.

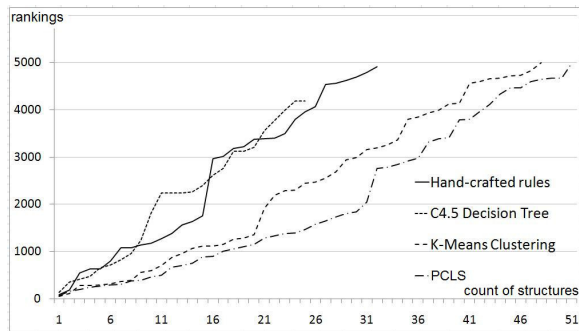


Fig. 3. Visualized results of extrinsic evaluation. It shows how the structures that actually have events are captured at the top 5,000 rank list. The vertical axis is the ranking, and the horizontal axis is a count of the number of structures. A lower curve is preferred, and the one more stretched to the right is preferred.

Table II summarizes the results of blind evaluation for Manhattan; larger scores for AUC and DCG, and lower for PNorm, correspond to superior performance. PCLS excels the other methods in all three measures. Figure 3 plots the vulnerable structures from the blind evaluation year (horizontal axis) against the top 5000 structures in the ranked lists from the four methods (vertical axis). As shown, PCLS outperforms the other methods in terms of all three measures (AUC, DCG, PNorm). Since the actual performance and scores are not linear correlated, even though PCLS is literally only 3.7% higher in AUC and 1.2% higher in DCG, the improvement is actually quite substantial. From Figure 3, in the top 5000 of the rank list, PCLS retrieves 51 vulnerable structures and C4.5 DT is 31, which is 64.5% improvement. When compared to KM, PCLS captures 3 more structures and the positions of these structures in the list are significantly ranked higher. Moreover, KM labels many more tickets as serious, as indicated by the very low precision and relatively high recall for this class (Table I), leading to much higher computational complexity of the structure ranking task for KM in contrast to PCLS.

## VIII. CONCLUSION

We have presented an electrical event categorization task on a power grid application system. The characteristics of the categorization problem require an approach that can adapt existing knowledge about the data model over time.

We developed a semi-supervised learning method, Progressive Clustering with Learned Seeds, that suited the problem. Intrinsic evaluation displays the stability and consistency of knowledge preservation in accordance with a set of very limited labeled data. Extrinsic evaluation shows the superior actual performance on a blind test. Our problem engineering method can also be implemented and adapted to other domains providing an exemplary approach that uses computational intelligence technology for industrial applications.

## REFERENCES

- [1] J. C. Schlimmer and R. H. Granger, "Incremental learning from noisy data," *Machine Learning*, vol. 1, no. 3, pp. 317–354, 1986.
- [2] P. E. Utgoff, "Incremental induction of decision trees," *Machine Learning*, vol. 4, no. 2, pp. 161–186, 1989.
- [3] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, Canada, 1990.
- [4] J. M. Pena, J. A. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the K-means algorithm," *Pattern Recognition Letters*, vol. 20, pp. 1027 – 1040, 1999.
- [5] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, June 2007, pp. 1027 – 1035.
- [6] P. S. Bradley and U. M. Fayyad, "Refining initial points for K-means clustering," in *Proceedings of the 15th International Conference on Machine Learning (ICML)*, 1998, pp. 91 – 99.
- [7] B. Liu, Y. Xia, and P. S. Yu, "Clustering through decision tree construction," in *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM)*, McLean, VA, 2000.
- [8] C. Rudin, R. J. Passonneau, A. Radeva, H. Dutta, S. Jerome, and D. Isaac, "A process for predicting manhole events in manhattan," *Mach. Learn.*, vol. 80, no. 1, pp. 1–31, Jul. 2009.
- [9] C. Rudin, "The P-Norm Push: A simple convex ranking algorithm that concentrates at the top of the list," *Journal of Machine Learning Research*, vol. 10, pp. 2233–2271, Oct 2009.
- [10] R. J. Passonneau, C. Rudin, A. Radeva, and Z. A. Liu, "Reducing noise in labels and features for a real world dataset: Application of nlp corpus annotation methods," in *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, 2009.
- [11] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, pp. 37–46, 1960.
- [12] R. Kittredge, "Sublanguages," *American Journal of Computational Linguistics*, pp. 79–84, 1982.
- [13] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Machine Learning Research*, pp. 1289–1305, 2003.
- [14] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA: Morgan Kaufmann, 2005.
- [15] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.
- [16] J. R. Galliers and K. Sparck Jones, "Evaluating natural language processing systems," Computer Laboratory, University of Cambridge, Tech. Rep. Technical Report 291, 1993.